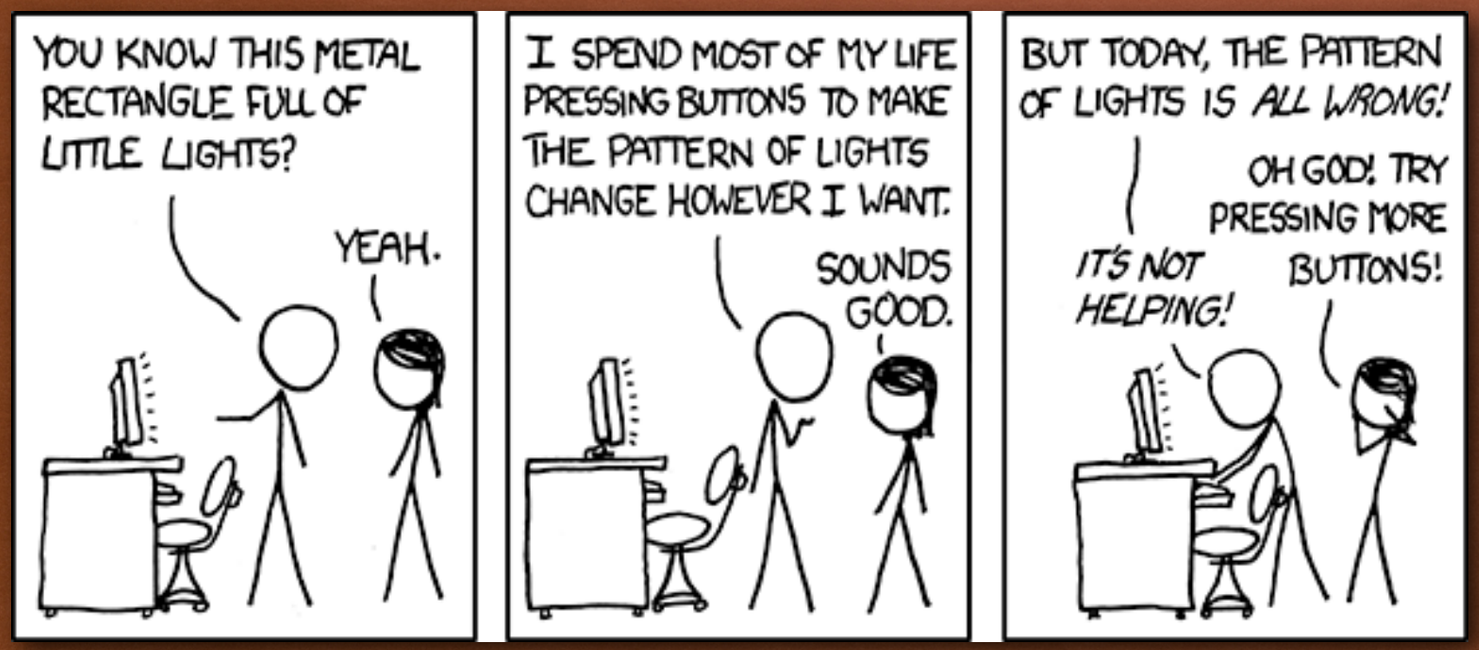
INTRODUCTION TO COMPUTER SYSTEMS

COMPSCI 110



https://xkcd.com/722/

COURSE LEARNING OUTCOMES

At the end of this course students should be able to:

- describe the layers of a computer system, from hardware to the web (1, 4)
- convert standard data types into numeric formats and apply simple functions to them (1)
- identify basic gates and be able to relate them to truth tables and simple combinational circuits (1, 2)
- trace the execution of simple programs at the assembly language level (1, 2)
- produce pseudocode to specify solutions to simple programming tasks (1, 2, 3)
- describe the fundamental parts of an operating system and how they relate to applications (1, 4)
- demonstrate an understanding of network communications and computer security (1, 4)
- explain the central ideas in computer science research areas (such as artificial intelligence, computer graphics and theory) (1, 4)
- discuss ethical issues arising from the use of computers in society (4, 5, 6)

WHAT IS THE COURSE ABOUT?

- We want to solve problems.
- We can use computers to solve problems.
  - To do this we need to convert our knowledge into simple data.
  - We also need to design and build machines (computers) which can manipulate that data; transforming the data and making decisions based on the data.
  - Then we need to make it easier for humans to use these machines - this leads to high-level programming languages.
  - By connecting the machines together we greatly increase the types of problems we can solve with them.
- That is what this course is about.

MAJOR THEME OF THE COURSE

- Abstraction
  - Different ways of viewing abstraction:
    - A generalised concept e.g. an abstract chair
    - Only concentrating on the parts of something which are relevant to the current task
    - Taking a different view of a layer supporting the one we are currently interested in
- So we reduce the relevant information so either make it easier to work with or to generalise

AN x64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND ITS GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND

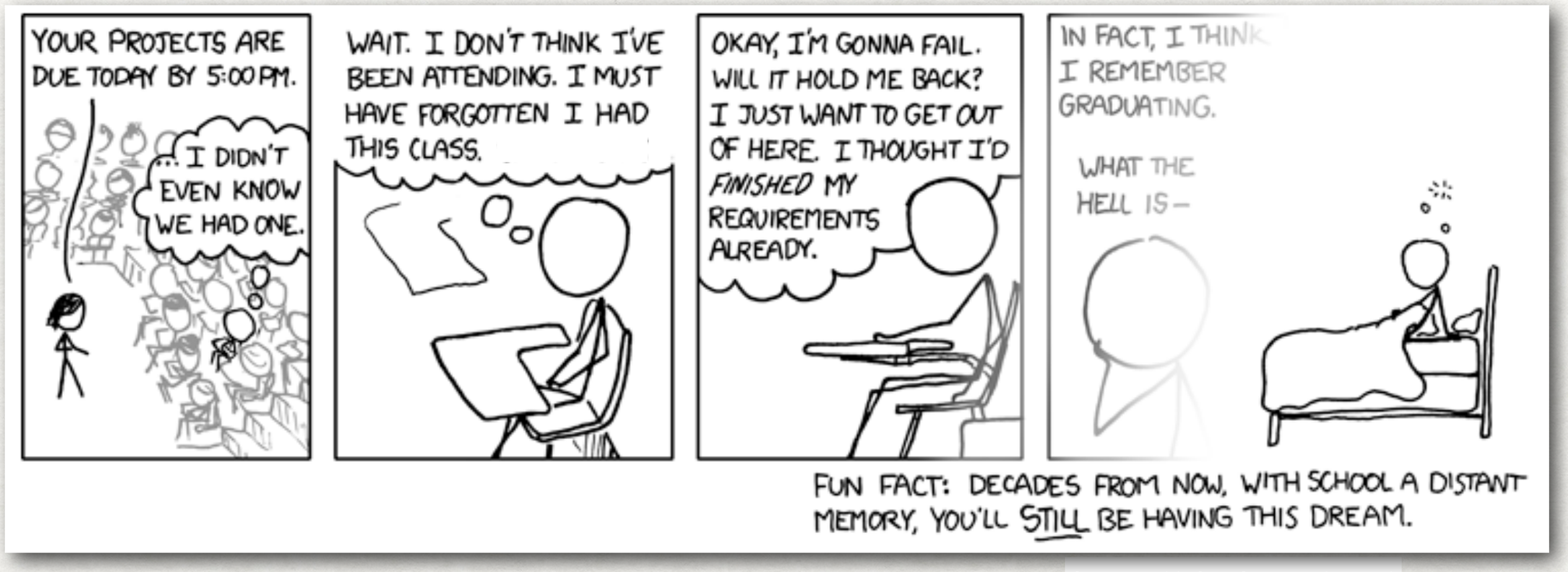
BECAUSE I WANTED TO SEE A CAT JUMP INTO A BOX AND FALL OVER.

I AM A GOD.

https://xkcd.com/676/

LECTURES AND TUTORIALS

- 3 lecture hours per week
- 1 tutorial per week for 10 weeks (starting this Thursday)
- 6 extra hours per week for you to do the essays and assignments, read the textbook, solve the textbook questions and revise the material



https://xkcd.com/557/

https://xkcd.com/557/

HOW DO I PASS?

- Hopefully because you learned some things 😀
- 5 Assignments 20% - 4% each
- Tutorial work 2% (0.2% each tutorial)
- 3 essays 8% - 2.666…% each
- 20% Test (Thursday the 23rd of September 6:30 - 7:30pm)
- 50% Exam (date to be determined, online, open book)

- Add it all up and you get a pass if you get at least 50% overall.

WE HAVE PEOPLE TO HELP

- Course coordinator - Dr Robert Sheehan - weeks 1-3
  - r.sheehan@auckland.ac.nz
  - My office hours: Tuesday and Thursday at 3pm (room 303.409)
- Other lecturers -
  - Dr Damir Azhar damir.azhar@auckland.ac.nz
  - Asma Shakil asma.shakil@auckland.ac.nz
  - Dr Thomas Lacombe thomas.lacombe@auckland.ac.nz
- Tutors -
  - Aditi Ramalingam aram485@aucklanduni.ac.nz
  - Alexander Swain aswa408@aucklanduni.ac.nz

WE HAVE A TEXTBOOK

- The textbook is the 8th edition of "Invitation to Computer Science" by Schneider and Gersting. The book is referenced throughout the course and you should get either an electronic or physical version of the book.
- You can get a physical version of the book from Ubiq price $157.46.
- To purchase the ebook version go to : nz.cengage.com
  - Type in "Invitation to Computer Science", it will come up with quite a number of versions, choose the 8th edition. The eBook price is $77.95 for five year access.
  - Please get the "Textbook/ebook" version for $77.95 NOT the Digital Platform version for $79.95.
  - Once you have selected the text then click checkout and apply discount code LEARN10NZ to receive 10% off your purchase.

OTHER RESOURCES

- The Piazza forum for questions about the course
  - students are encouraged to answer these, your postings can be anonymous to the class
  - tutors and lecturers will also keep an eye and help out
- Previous tests and exams
  - We make some of these available and provide answers for some (but not all).
  - Please don’t get upset when we don’t provide all answers for previous tests and exams
    - We are happy for the class to generate solutions to entire tests
    - Discussing answers is a great way to learn

AN INTRODUCTION TO COMPUTER SCIENCE

CHAPTER 1

WHAT IS COMPUTER SCIENCE?

- Which of these do you think best sums up what Computer Science is?
  - the study of computers
  - the study of how to write computer programs
  - the study of the uses and applications of computers and software
  - the study of algorithms

- https://www.youtube.com/watch?v=fjMU-km-Cso&list=PLbg3ZX2pWlgI_ej6ZhGd45-cPoWLZD9pT

THE TEXTBOOK DEFINITION OF COMPUTER SCIENCE

- Computer science is the study of algorithms, including:
  - Their formal and mathematical properties
  - Their hardware realizations
  - Their linguistic realizations
  - Their applications

WHAT IS COMPUTER SYSTEMS?

- A focus on the hardware end of computers and the software provided to make them usable.
  - So processors, memory, storage devices
  - but also operating systems, assemblers, compilers, security and networking
- Follow up (and related courses) - PHYSICS 140, COMPSCI 210, 215, 313, 315, 316, 340.

ALGORITHMS

- The Importance of Algorithmic Problem Solving
  - "Industrial revolution" of the nineteenth century
    - Mechanized and automated repetitive physical tasks
  - "Computer revolution" of the twentieth and twenty-first centuries
    - Mechanized and automated repetitive mental tasks
    - Used algorithms and computer hardware

https://www.youtube.com/watch?v=0OyhASRR0To (6:30 on)

CHARLES BABBAGE AND ADA LOVELACE

- Difference Engine designed and built in 1823
  - Could do addition, subtraction, multiplication, and division to six significant digits
  - Could solve polynomial equations and other complex mathematical problems
- Analytical Engine
  - Designed but never built
  - Mechanical, programmable machine with parts that mirror that of a modern-day computer:
    - Mill: Arithmetic/logic unit
    - Store: Memory
    - Operator: Processor
    - Output Unit: Input/Output

Ada Lovelace

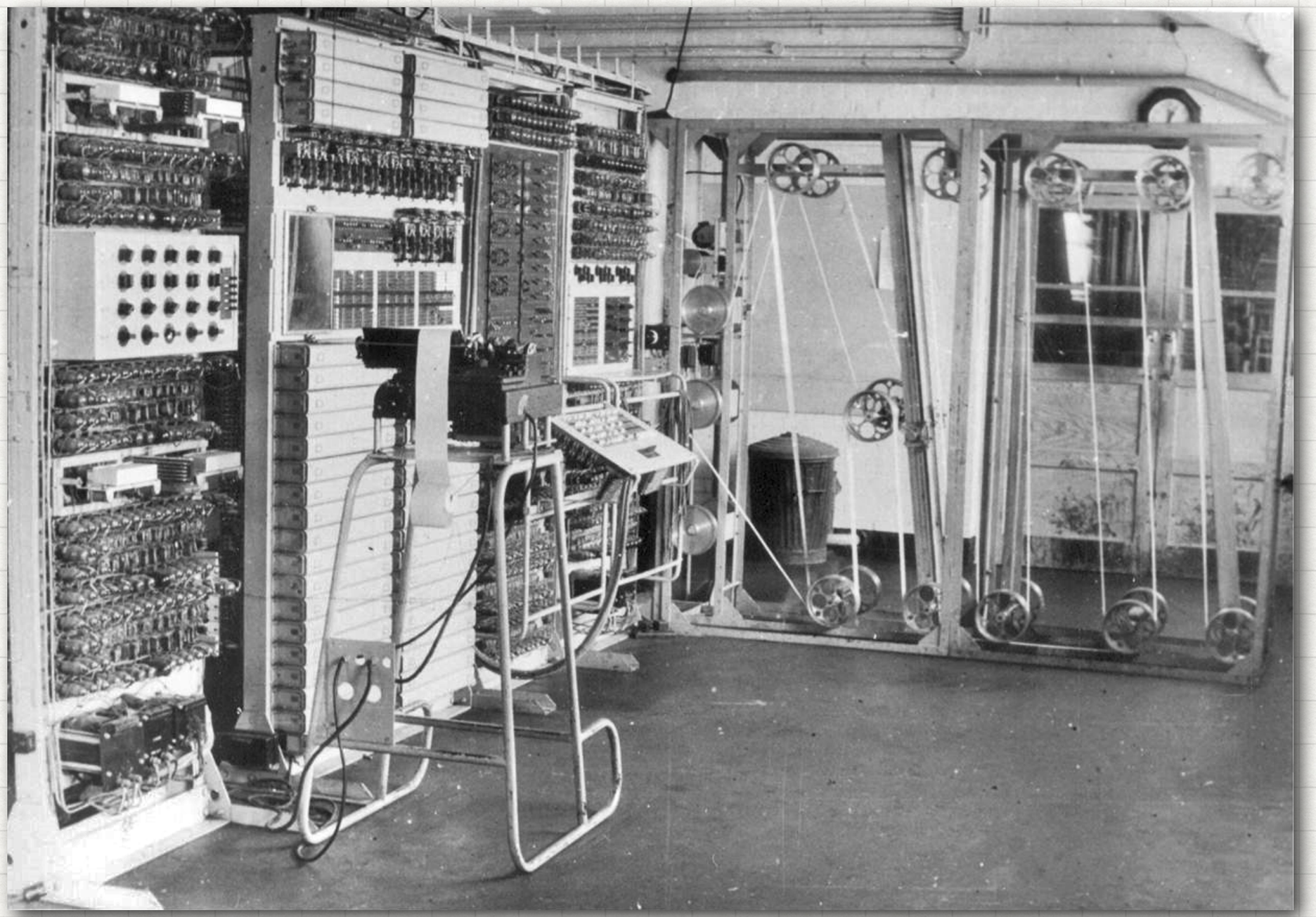
Wrote a book explaining programming the analytical engine

but also recognised its more general use

COMPUTERS BECOME REALISTIC WITH THE DEVELOPMENT OF ELECTRONICS

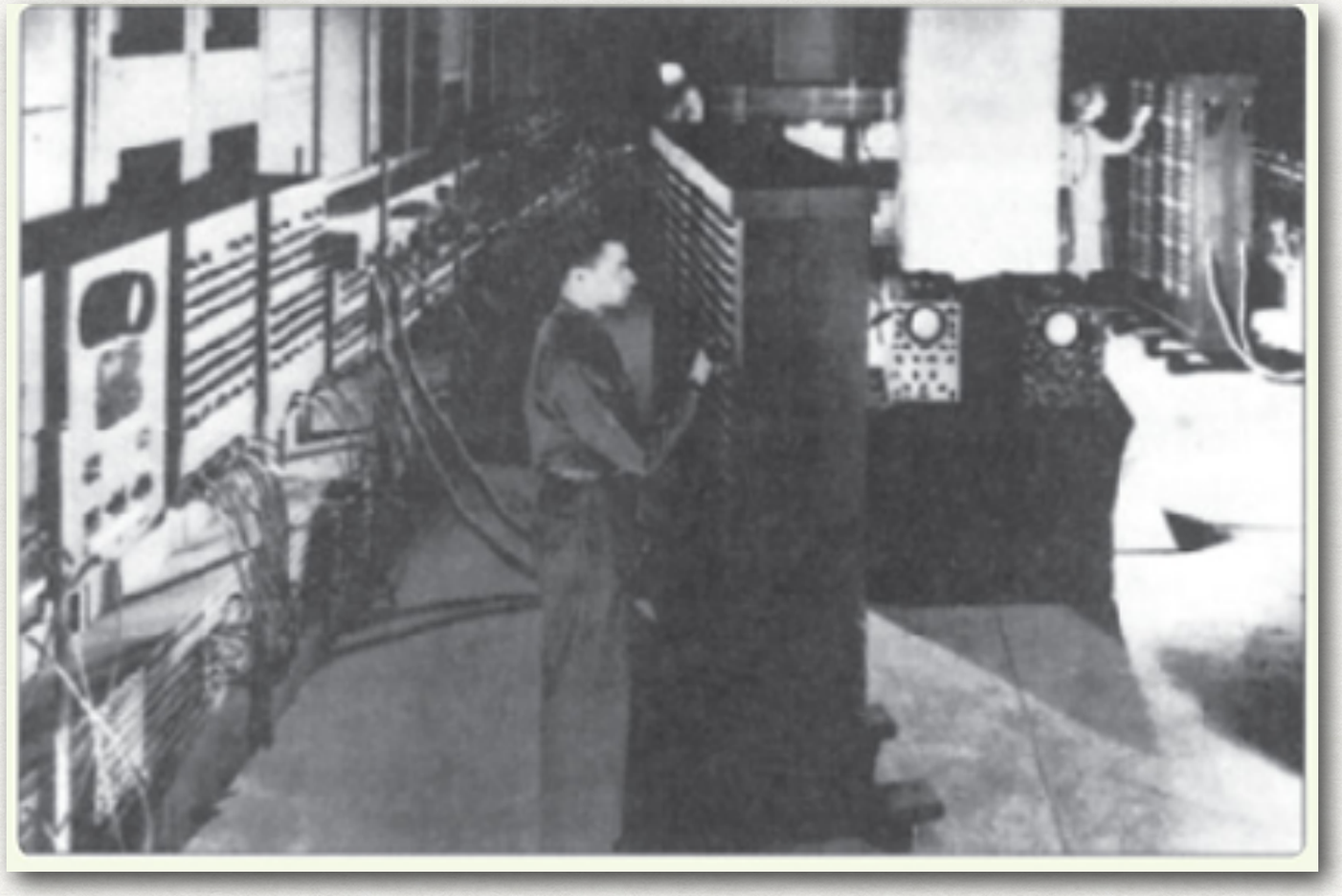
- Mark I (1944)
  - Electromechanical computer used a mix of relays, magnets, and gears to process and store data
- Colossus (1943)
  - General-purpose computers designed and built by Tommy Flower with input from Alan Turing for the British Lorenz cipher project (not Enigma)
- ENIAC (Electronic Numerical Integrator and Calculator) (1946)
  - First publicly known fully electronic computer
- Z1 (binary mechanical computer) (1938)
  - Konrad Zuse
- EDSAC (first electronic stored program computer?) (1949)
  - Maurice Wilkes (Cambridge University)

COLOSSUS



https://upload.wikimedia.org/wikipedia/commons/0/08/Wartime_photo_of_Colossus_10.png

FIGURE 1.7 PHOTOGRAPH OF THE ENIAC COMPUTER



Source: From the Collections of the University of Pennsylvania Archives (U.S Army photo)

JOHN VON NEUMANN

- Proposed a radically different computer design based on a model called the **stored program computer**
- Research group at the University of Pennsylvania built one of the first stored program computers, called EDVAC, in 1949
- UNIVAC I, a version of EDVAC, the first commercially sold computer
- Nearly all modern computers use the **Von Neumann architecture**